IN THE CLAIMS:

Please re-write the claims to read as follows:

1. (Previously Presented) A method for striping packets across pipelines of a processing engine within a network switch, the processing engine having a plurality of processors arrayed as pipeline rows and columns embedded between input and output buffers, the method comprising the steps of:

including a context memory in each pipeline row;

organizing the context memory as a plurality of window buffers of a defined size;

apportioning each packet into contexts corresponding to the defined size associated with each window buffer; and

correlating each context with a relative position within the packet to thereby facilitate reassembly of the packet at the output buffer, while obviating out-of-order issues involving the contexts of the packet.

2. (Original) The method of Claim 1 further comprising the step of organizing the processors and context memory of each pipeline row as a cluster.

3. (Original) The method of Claim 2 wherein the step of apportioning comprises the steps of:

segmenting the packet into fixed sized contexts at the input buffer;

sequentially passing the contexts to the clusters; and

storing the fixed sized contexts in appropriate window buffers of the context memories.

4. (Original) The method of Claim 3 wherein the step of correlating comprises the step of providing a program counter entry point function to indicate the relative position of each context within the packet.

5. (Original) The method of Claim 3 wherein the relative position comprises one of a beginning, middle and end context of the packet.

6. (Original) The method of Claim 3 further comprising the steps of:

processing the context at a source processor of the cluster;

communicating an intermediate result relating to processing of the context to a destination processor of a neighboring cluster.

7. (Original) The method of Claim 6 wherein the step of communicating comprises the step of providing an intercolumn communication mechanism configured to

forward the intermediate result from the source processor to an address of the destination processor.

8. (Original) The method of Claim 3 further comprising the step of changing the size of a fixed sized context at the context memory of a cluster.

9. (Previously presented) A method for striping packets across pipelines of a processing engine within a network switch, the processing engine having a plurality of processors arrayed as pipeline rows and columns embedded between input and output buffers, the method comprising the steps of:

including a context memory in each pipeline row;

organizing the context memory as a plurality of window buffers of a defined size;

apportioning each packet into contexts corresponding to the defined size associated with each window buffer by,

segmenting the packet into fixed sized contexts at the input buffer;

sequentially passing the contexts to the clusters; and

storing the fixed sized contexts in appropriate window buffers of the context memories;

correlating each context with a relative position within the packet to thereby facilitate reassembly of the packet at the output buffer, while obviating out-of-order issues involving the contexts of the packet;

organizing the processors and context memory of each pipeline row as a cluster;

changing the size of a fixed sized context at the context memory of a cluster;

deleting a portion of the fixed sized context stored in the window buffer; and

substituting the deleted portion of the context with information stored at another location of the context memory.

10. (Original) The method of Claim 9 wherein the substituted information is one of larger than and smaller than the deleted portion of the fixed sized context.

11. (Original) A system for striping packets across pipelines of a processing engine within a network switch, the processing engine having a plurality of processors arrayed as pipeline rows and columns embedded between input and output buffers, the system comprising:

a context memory within each pipeline row, the context memory organized as a plurality of window buffers of a defined size;

a segmentation unit adapted to apportion each packet into contexts for processing by the processors, each context corresponding to the defined size associated with each window buffer; and

a mapping mechanism configured to correlate each context with a relative position within the packet to thereby facilitate reassembly of the packet at the output buffer, while obviating out-of-order issues involving the contexts of the packet.

12. (Original) The system of Claim 11 wherein the processors and context memory of each pipeline row are organized as a cluster.

13. (Original) The system of Claim 12 wherein the mapping mechanism comprises a program counter entry point function that indicates the relative position of each context within the packet.

14. (Original) The system of Claim 13 wherein the relative position comprises one of a first, last and intermediate portion of the packet.

15. (Original) The system of Claim 13 further comprising an intercolumn communication mechanism configured to forward an intermediate result relating to processing of a context by a source processor to a destination processor.

16. (Original) A computer readable medium containing executable program instructions for striping packets across pipelines of a processing engine within a network switch, the processing engine having a plurality of processors arrayed as pipeline rows and columns embedded between input and output buffers, each pipeline row including a context memory, the processors and context memory of each pipeline row organized as a cluster, the executable program instructions comprising program instructions for:

organizing the context memory as a plurality of window buffers of a defined size;

apportioning each packet into contexts corresponding to the defined size associated with each window buffer; and

correlating each context with a relative position within the packet to thereby facilitate reassembly of the packet at the output buffer, while obviating out-of-order issues involving the contexts of the packet.

17. (Original) The computer readable medium of Claim 16 further comprising program instructions for:

segmenting the packet into fixed sized contexts at the input buffer;

sequentially passing the contexts to the clusters; and

storing the fixed sized contexts in appropriate window buffers of the context memories.

18. (Original) The computer readable medium of Claim 17 wherein the program instruction for correlating comprises the program instruction for providing a program counter entry point function to indicate the relative position of each context within the packet.

19. (Original) The computer readable medium of Claim 17 further comprising program instructions for changing the size of a fixed sized context at the context memory of a cluster.

20. (Previously Presented) A computer readable medium containing executable program instructions for striping packets across pipelines of a processing engine within a network switch, the processing engine having a plurality of processors arrayed as pipeline rows and columns embedded between input and output buffers, each pipeline row including a context memory, the processors and context memory of each pipeline row organized as a cluster, the executable program instructions comprising program instructions for:

organizing the context memory as a plurality of window buffers of a defined size;

apportioning each packet into contexts corresponding to the defined size associated with each window buffer;

correlating each context with a relative position within the packet to thereby facilitate reassembly of the packet at the output buffer, while obviating out-of-order issues involving the contexts of the packet;

segmenting the packet into fixed sized contexts at the input buffer;

sequentially passing the contexts to the clusters;

storing the fixed sized contexts in appropriate window buffers of the context memories;

changing the size of a fixed sized context at the context memory of a cluster;

deleting a portion of the fixed sized context stored in the window buffer; and

substituting the deleted portion of the context with information stored at another location of the context memory.

21. (Previously Presented) Electromagnetic signals propagating on a computer network carrying instructions for striping packets across pipelines of a processing engine within a network switch, the processing engine having a plurality of processors arrayed as pipeline rows and columns embedded between input and output buffers, each pipeline row including a context memory, the processors and context memory of each pipeline row organized as a cluster, the electromagnetic signals comprising program instructions for:

organizing the context memory as a plurality of window buffers of a defined size;

apportioning each packet into contexts corresponding to the defined size associated with each window buffer; and

correlating each context with a relative position within the packet to thereby facilitate reassembly of the packet at the output buffer, while obviating out-of-order issues involving the contexts of the packet.

22. (Previously Presented) A method for operating a network switch, comprising:

arraying a plurality of processors as a plurality of rows, a row forming a pipeline row, the pipeline rows arrayed between an input buffer and an output buffer;

including a context memory in each pipeline row;

organizing the context memory as a plurality of window buffers of a defined size;

apportioning, by the input buffer, each packet into packet contexts, a packet context corresponding to the defined size associated with each window buffer; and

correlating each packet context with a relative position within the packet to thereby facilitate reassembly of the packet at the output buffer, to facilitates striping packets across a plurality of the pipelines.

23. (Previously Presented) The method of claim 22, further comprising:

organizing the processing engine so that a pipeline row forms a cluster of processors, and including the context memory as part of the cluster.

24. (Previously Presented) The method of claim 22, further comprising:

segmenting the packet into fixed sized contexts at the input buffer;

sequentially passing the contexts to the clusters; and

storing the fixed sized contexts in appropriate window buffers of the context memories.

25. (Previously Presented) The method of claim 22, further comprising:

providing a program counter entry point function to indicate the relative position of each context within the packet.

26. (Previously Presented) The method of claim 22, wherein the relative position further comprises:

one of a beginning, middle and end context of the packet.

27. (Previously Presented) The method of claim 22, further comprising:

processing the context at a source processor of the cluster;

communicating an intermediate result relating to processing of the context to a destination processor of a neighboring cluster.

28. (Previously Presented) The method of claim 27, further comprising:

providing an intercolumn communication mechanism configured to forward the intermediate result from the source processor to an address of the destination processor.

29. (Previously Presented) The method of claim 22, further comprising:

changing the size of a fixed sized context at the context memory of a cluster.

30. (Currently Amended) ~~The method of claim 22, further comprising:~~

A method for operating a network switch, comprising:

arraying a plurality of processors as a plurality of rows, a row forming a pipeline row, the pipeline rows arrayed between an input buffer and an output buffer;

including a context memory in each pipeline row;

organizing the context memory as a plurality of window buffers of a defined size;

apportioning, by the input buffer, each packet into packet contexts, a packet context corresponding to the defined size associated with each window buffer; correlating each packet context with a relative position within the packet to thereby facilitate reassembly of the packet at the output buffer,

deleting a portion of the fixed sized context stored in the window buffer; and

substituting the deleted portion of the context with information stored at another location of the context memory.

31. (Previously Presented) A processing engine within a network switch, comprising:

means for arraying a plurality of processors as a row of a plurality of pipelines, the rows arrayed between an input buffer and an output buffer;

means for including a context memory in each pipeline row;

means for organizing the context memory as a plurality of window buffers of a defined size;

means for apportioning, by the input buffer, each packet into packet contexts, a packet context corresponding to the defined size associated with each window buffer; and

means for correlating each packet context with a relative position within the packet, to facilitate reassembly of the packet at the output buffer, thereby facilitating striping packets across the plurality of pipelines.

32. (Previously Presented) The processing engine of claim 31, further comprising:

means for organizing the processing engine so that a pipeline row forms a cluster of processors, and including the context memory as part of the cluster.

33. (Previously Presented) The processing engine of claim 31, further comprising:

means for segmenting the packet into fixed sized contexts at the input buffer;

means for sequentially passing the contexts to the clusters; and

means for storing the fixed sized contexts in appropriate window buffers of the context memories.

34. (Previously Presented) The processing engine of claim 31, further comprising:

means for providing a program counter entry point function to indicate the relative position of each context within the packet.

35. (Previously Presented) The processing engine of claim 31, wherein the relative position further comprises:

means for one of a beginning, middle and end context of the packet.

36. (Previously Presented) The processing engine of claim 31, further comprising:

means for processing the context at a source processor of the cluster;

means for communicating an intermediate result relating to processing of the context to a destination processor of a neighboring cluster.

37. (Previously Presented) The processing engine of claim 36, further comprising:

means for providing an intercolumn communication mechanism configured to forward the intermediate result from the source processor to an address of the destination processor.

38. (Previously Presented) The processing engine of claim 31, further comprising:

means for changing the size of a fixed sized context at the context memory of a cluster.

39. (Currently Amended) ~~The processing engine of claim 31, further comprising:~~

A processing engine within a network switch, comprising:

means for arraying a plurality of processors as a row of a plurality of pipelines, the rows arrayed between an input buffer and an output buffer;

means for including a context memory in each pipeline row;

means for organizing the context memory as a plurality of window buffers of a defined size;

means for apportioning, by the input buffer, each packet into packet contexts, a packet context corresponding to the defined size associated with each window buffer;

means for correlating each packet context with a relative position within the packet, to facilitate reassembly of the packet at the output buffer;

means for deleting a portion of the fixed sized context stored in the window buffer; and

means for substituting the deleted portion of the context with information stored at another location of the context memory.

40. (Previously Presented) A processing engine within a network switch, comprising:

a plurality of processors arrayed as a plurality rows, a row forming a pipeline, the plurality of processors forming a plurality of rows of pipelines, the rows arrayed between an input buffer and an output buffer;

a context memory included in each pipeline row;

the context memory organized as a plurality of window buffers of a defined size;

the input buffer apportioning each packet into packet contexts, a packet context corresponding to the defined size associated with each window buffer; and

a processor of the plurality of processors to correlate each packet context with a relative position within the packet, to facilitate reassembly of the packet at the output buffer, thereby facilitating striping packets across the plurality of pipelines.

41. (Previously Presented) The processing engine of claim 40, further comprising:

the processing engine organized so that a pipeline row forms a cluster of processors, and including the context memory as part of the cluster.

42. (Previously Presented) The processing engine of claim 40, further comprising:

the input buffer segmenting the packet into fixed sized contexts;

the input buffer sequentially passing the contexts to the clusters; and

window buffers to store the fixed sized contexts in appropriate context memories.

43. (Previously Presented) The processing engine of claim 40, further comprising:

a program counter entry point function to indicate the relative position of each context within the packet.

44. (Previously Presented) The processing engine of claim 40, wherein the relative position further comprises:

a beginning, middle and end context of the packet.

45. (Previously Presented) The processing engine of claim 40, further comprising:

a source processor of the cluster to process the context; and

a destination processor of a neighboring cluster to receive an intermediate result relating to processing of the context.

46. (Previously Presented) The processing engine of claim 45, further comprising:

an intercolumn communication mechanism configured to forward the intermediate result from the source processor to an address of the destination processor.

47. (Previously Presented) The processing engine of claim 40, further comprising:

a processor of the plurality of processors to change the size of a fixed sized context at the context memory of a cluster.

48. (Previously Presented) The processing engine of claim 40, further comprising:

a processor of the plurality of processors to change the size of a fixed sized context at the context memory of a cluster.